```
/*
TinyTrackGPS.ino
v0.2

Copyright © 2019-2021 Francisco Rafael Reyes Carmona.
All rights reserved.

rafael.reyes.carmona@gmail.com

  This file is part of TinyTrackGPS.

  TinyTrackGPS is free software: you can redistribute it and/or modify
  it under the terms of the GNU General Public License as published by
  the Free Software Foundation, either version 3 of the License, or
  (at your option) any later version.

  TinyTrackGPS is distributed in the hope that it will be useful,
  but WITHOUT ANY WARRANTY; without even the implied warranty of
  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
  GNU General Public License for more details.

  You should have received a copy of the GNU General Public License
  along with TinyTrackGPS.  If not, see <https://www.gnu.org/licenses/>.
*/

/*
     Programa de localizacion por gps que graba las posiciones en
     un fichero de texto cada segundo, de forma diaria.

     Conectar módulo SD con pin CS (naranja) en pin 10 arduino.

     Conectar módulo NMEA-6M (gps) pines 8,9 (9 - pin rx negro)

     Conectar LCD 16x2 pines 2,3,4,5,6,7 (2-amarillo , 3-azul,
     4-rojo, 5-azul oscuro, 6-verde, 7-blanco)
*/
// Include libraries.
#include <LiquidCrystal.h>
#include <SoftwareSerial.h>
#include <TinyGPS.h>
#include <SD.h>

File GPSFile;
char GPSLogFile[] = "YYYYMMDD.csv"; // Formato de nombre de fichero. YYYY-
Año, MM-Mes, DD-Día.
float flat, flon;
unsigned long age;

/* Código de demostración uso de librería TinyGPS.
   Requiere uso de librería SoftwareSerial, se presupone que disponemos
   de un dispositivo GPS serie de 9600-bauds conectado en pines 9(rx) y
```

```
    de un dispositivo GPS serie de 9600 bauds conectado en pines 9(rx) y
    8(tx).
*/
TinyGPS gps;
SoftwareSerial gps_serial(9, 8);
int year_actual;
byte month_actual, day_actual;

/* Constantes para declaracion del LCD */
const int LCD_NB_ROWS = 2;
const int LCD_NB_COLUMNS = 16;
/* Crea el objeto lcd */
LiquidCrystal lcd(2, 3, 4, 5, 6, 7);

void setup(void) {
  bool config = false;
  Serial.begin(9600);
  gps_serial.begin(9600);

  pinMode(10, OUTPUT);

  Serial.print(F("Initializing SD card..."));

  if (!SD.begin(10)) {
    Serial.println(F("FAILED!"));
    return;
  }
  Serial.println(F("Done."));

  /* Iniciaización del LCD */
  lcd.begin(LCD_NB_COLUMNS, LCD_NB_ROWS);
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print(F("Waiting for"));
  lcd.setCursor(0, 1);
  lcd.print(F("GPS signal."));

  do {
  while (gps_serial.available())
    {
      if (gps.encode(gps_serial.read())) // Comprueba que ha recibido una
      sentencia del GPS.
        {
          int year;
          byte month, day, hour, minute, second, hundredths;
          unsigned long age;

          gps.crack_datetime(&year, &month, &day, &hour, &minute, &second,
          &hundredths, &age);
          if (sprintf(GPSLogFile, "%04d%02d%02d.csv", year, month, day) >
          0) config = true;
```

```
97            year_actual = year;
98            month_actual = month;
99            day_actual = day;
100
101           Serial.print(F("Filename: "));
102           Serial.println(GPSLogFile);
103
104           // Si no existe el fichero lo crea y añade las cabeceras.
105           if (!SD.exists(GPSLogFile)) {
106              if (GPSFile = SD.open(GPSLogFile, FILE_WRITE)) {
107                Serial.print(F("New GPSLogFile, adding heads..."));
108                GPSFile.println(F("Time,latitude,longitude"));
109                Serial.println(F("Done."));
110                GPSFile.close();
111             } else {
112                Serial.println(F("** Error creating GPSLogFile. **"));
113             }
114           }
115         }
116     }
117   }while(!config);
118 }
119
120 void loop(void) {
121 //  unsigned long chars;
122 //  unsigned short sentences, failed;
123   int year;
124   byte month, day, hour, minute, second, hundredths;
125   unsigned long age;
126   char timestr[10];
127
128   // For one second we parse GPS data and report some key values
129   for (unsigned long start = millis(); millis() - start < 1000;)
130   {
131     while (gps_serial.available())
132     {
133       if (gps.encode(gps_serial.read())) { // Did a new valid sentence
      •       come in?
134         gps.f_get_position(&flat, &flon, &age);
135         lcd.setCursor(0, 0);
136         lcd.print(F("LAT="));
137         lcd.print(flat == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : flat, 6);
138         lcd.setCursor(0, 1);
139         lcd.print(F("LON="));
140         lcd.print(flon == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : flon, 6);
141 /*
142         gps.stats(&chars, &sentences, &failed);
143         Serial.print(" CHARS=");
144         Serial.print(chars);
145         Serial.print(" SENTENCES=");
146         Serial.print(sentences);
```

```
        Serial.print(" CSUM ERR=");
        Serial.println(failed);
        if (chars == 0)
          Serial.println("** No characters received from GPS: check wiring
          **");
*/

        gps.crack_datetime(&year, &month, &day, &hour, &minute, &second,
        &hundredths, &age);
        if (age != TinyGPS::GPS_INVALID_AGE){
          sprintf(timestr, "%02d:%02d:%02d,", hour, minute, second);
        }
        if (year != year_actual || month != month_actual || day !=
        day_actual) {
          sprintf(GPSLogFile, "%04d%02d%02d.csv", year, month, day);
          year_actual = year;
          month_actual = month;
          day_actual = day;

          // Si no existe el fichero lo crea y añade las cabeceras.
          if (!SD.exists(GPSLogFile)) {
             if (GPSFile = SD.open(GPSLogFile, FILE_WRITE)) {
               Serial.print(F("New GPSLogFile, adding heads..."));
               GPSFile.println(F("Time,latitude,longitude"));
               Serial.println(F("Done."));
               GPSFile.close();
             } else {
               Serial.println(F("** Error creating GPSLogFile. **"));
             }
          }
        }
        if (GPSFile = SD.open(GPSLogFile, FILE_WRITE)) {
          Serial.print(F("Open GPSLogFile to write..."));
          GPSFile.print(timestr);
          GPSFile.print(flat,6);
          GPSFile.print(",");
          GPSFile.println(flon,6);
          GPSFile.close();
          Serial.println(F("Done."));
        } else {
          Serial.println(F("** Error opening GPSLogFile. **"));
        }
      }
    }
  }
}
```